

Magento 设计人员专业名词（Magento Design Terminologies）

为了能够更好的理解本设计人员中文手册，熟悉一下 Magento 系统中的设计方面的术语是很关键的。术语的这一章介绍可能对于你来说是比较新的概念，因此，您可以利用这段时间彻底的读通他们。但最重要的，如果你不能充分把握这些概念，不要灰心，所有这些新名词-本章仅仅向你介绍一次，在后面的章节会对这些简单的定义进一步深入研究和扩大。这一章介绍的术语包括：

网站和网店（Website and Store）

Magento 中的一个网站（Website）是一些共享相同的客户、订单信息和购物车信息的商店（Store）的集合。Store 是一些 Store View 的集合。这些都是比较笼统的概念，我们在确立个别特别需要的网店时用这些概念可以很好的建立。下面有几种确定不同 website, store 和 store views 用途的：

方案 1



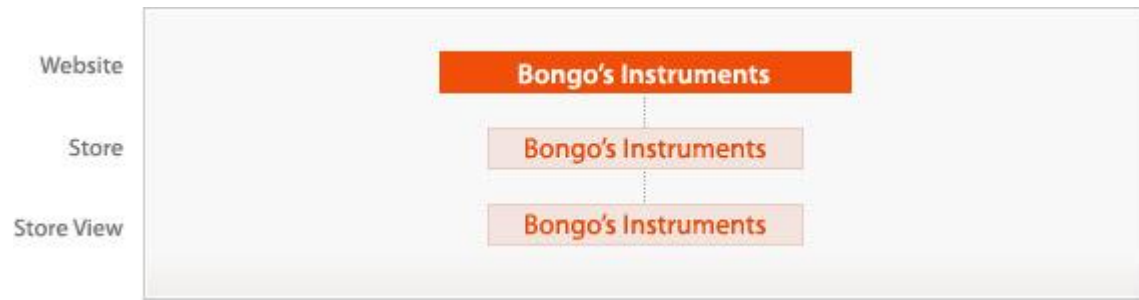
一家名为 Dubloo 的公司创建了一个在线销售平台，它包括三个独立的服装商店，每个迎合不同价格水平的受众。Dubloo 公司希望三个商店能够共享所有客户和订单信息。在这种情况下，Dubloo 公司将有一个网站（Website），下面有三个三个商店（Store）。商店将确定个别的价格水平商店，网站将成为 Dubloo 公司的总平台。

方案 2



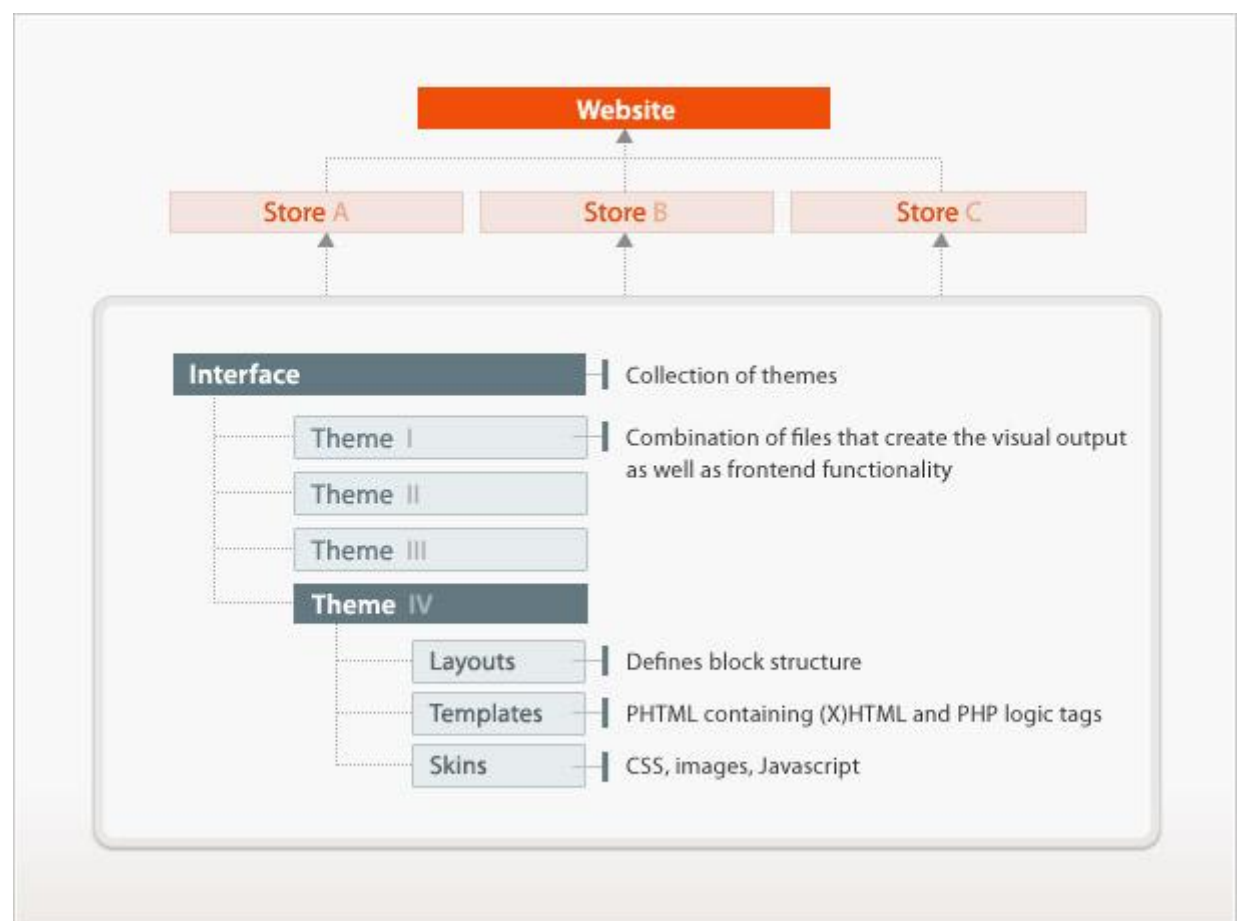
一家名为 My Laptops 要建立两个单独的网站，两个都是销售笔记本电脑，但是不同的价格。他们还希望每个站点提供英文和西班牙文的选择，每个客户可以在语言选择中选择自己的语言。它们还需要两个网站同步客户和订单信息。在这种情况下，Store View 将在网站中将确定每个英文和西班牙文版本。两个网站分别是“My Laptops”和“Cheap Laptops”。

方案 3



一家名为 Bongo's Instruments 想创建一个在线的销售网点。没有其他分支店，Bongo's Instruments 就是是商店以及网站。

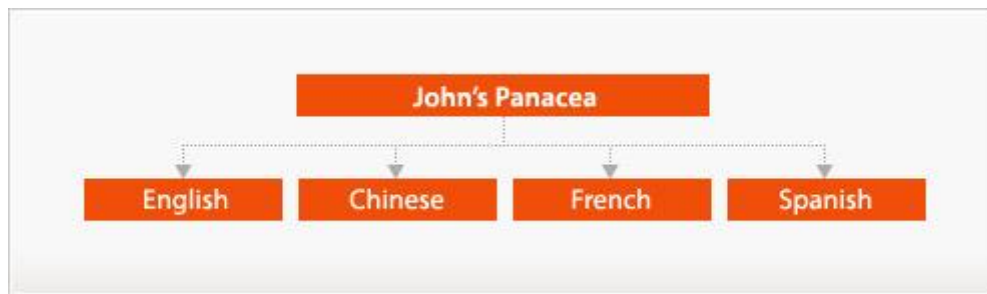
界面（Interface）



Magento 的界面(Interface)是一些主题(Themes)的集合,主题是确定网店的外观和前台页面结构的。一个界面可以在 Magento 后台为网站级别和/或 Store View 级别指定（了解如何为网站/商店分配一个界面）。

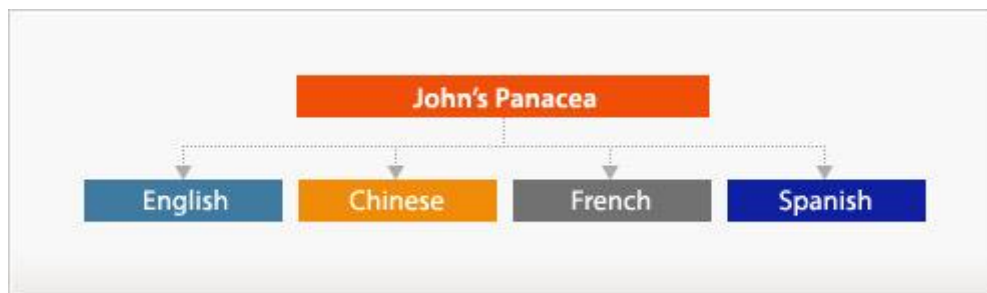
如果您指派一个在网站级别的界面，所有在这个网站下的商店将继承此界面。你可以进一步指定一个在 Store View 和商店级别的界面，可以有效的覆盖继承来的网站的界面。例如你在一个名为“John's Pancea”的网站下经营四个不同的商店-通过学习下面每个方法的效果，您可以轻松地根据你的业务的需求来确定部署设计的方法。

1. 网站级别的声明（Website-level declaration）



如果您想为四个商店都建立一个统一的外观和感觉的设计，您将指定一个网站级别的界面，在这种情况下，所有四个商店将继承网站的界面。

2. Store View 级别的声明（Store view-level declaration）



如果你想每家商店都有一个单独的外观和感觉的设计，你可以为每个 Store View 指定一个界面，在这种情况下，四个商店都有其独特的外观和感觉。

主题（Themes）

Magento 中的一个主题（Theme）是由布局（Layout），模板（Template）和皮肤文件（skin 文件是控制网店的显示效果的）组成。Magento 被设计成一次可以加载多个主题的功能，因此，通过两种类型来区分主题：

- **默认主题（Default theme）**

每个界面都有一个主要的主题，称之为界面的默认主题。当你为你的网店指定一个界面的时候，系统就会自动地寻找这个默认主题，并按照这个主题来显示前台。为了自定义网店设计，您可以修改这个主题，或新建一个非默认主题，并加载它。默认的主题必须包含所有必要的布局，模板和皮肤，以确保网店顺利运行，作为主题层次系统中最低级的主题。

- **非默认主题（Non-default theme）**

非默认主题可以根据你的需要来包含或多或少的主题元素文件（Layout、Template 和 Skin）。这种类型的主题可以作为网店的临时季节性的更改，而没有必要建立一个新的默认主题，通常可以通过创建的几张图片和更新一些 CSS 来实现，例如：你可以轻松地让您的网店从平常的页面外观变成有圣诞节气氛的网店。

让我们来了解一些主题的组件：

- **布局 (Layout)** (在 `app/design/frontend/your_interface/your_theme/layout/`)
布局是 XML 文件，它的作用是定义不同页面的区块 (Block) 结构，以及控制页面的 META 信息和网页的编码。布局文件是基于每个模块分布的，每一个模块 拥有它自己的布局文件。要深入了解 Layout，请阅读布局的介绍。
- **模板 (Templates)** (在 `app/design/frontend/your_interface/your_theme/template/`)
模板是 PHTML 文件，它包含 (X) HTML 标签和一些用来实现信息和功能显示的逻辑的 PHP 代码。
- **本地 (Locale)** (在 `app/design/frontend/your_interface/your_theme/locale/`)
这些都是在每个语言基础上组织的简单的文本文件，包含商店的翻译副本。
- **皮肤 (Skins)** (在 `skin/frontend/your_interface/your_theme/`)
皮肤是具体区块 (Blocks) 中的 JavaScript、CSS 和图片文件。你可能会问什么是区块？很好的问题，不过不用担心，Magento 已经为这个 组件定义好了，看下一节。

区块 (Blocks)

图 1. 结构区块(Structural Block) (蓝色区域)

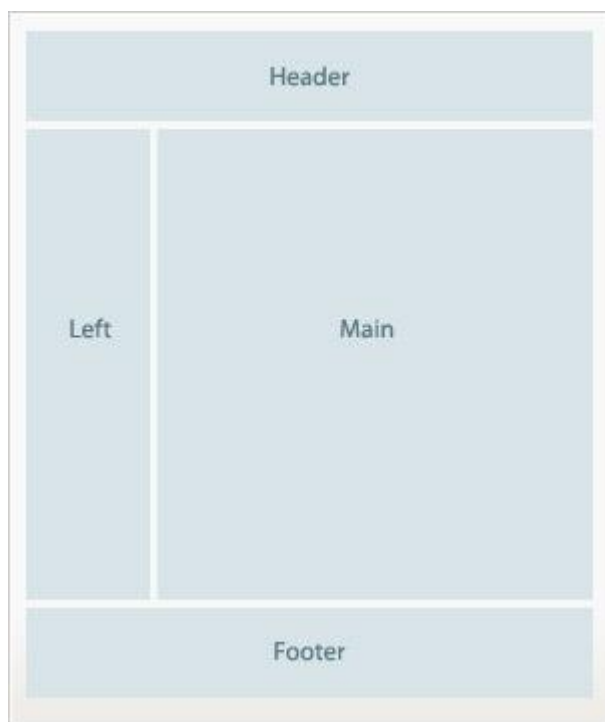
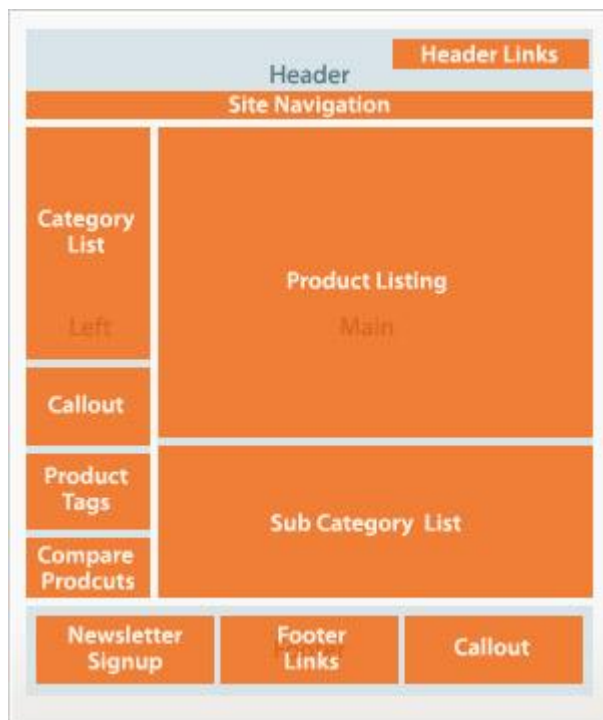


图 2. 内容区块（Content Block）（橙色区域）



区块（Block）是 Magento 用来区分系统中一系列功能，并创建一个可见的、操作方式统一的模块化方式来管理这些功能的方式。有两种类型的 Block，并且它们相互工作来创建输出的内容：

- **结构区块（Structural Blocks）**

这些 Blocks 是设计成来创建网店页面的可视结构的。例如：头部（Header）、左边栏（Left Column）、主体内容（Main Column）和页眉（Footer），看图 1。

- **内容区块（Content Blocks）**

这些 Blocks 是在 Structural Block 中生成实际的内容的。它们是每个功能的表现块，它们会部署 Template Files 到 Structural Block 中生成(X)HTML 内容。例如：分类列表（Category List），小购物车（Mini Cart），产品标签（Product Tags）和产品列表（Product Listing）等等就是它们自己的内容区块，见图 2。

模板中包含模板以组织全体的(X)HTML 输入是作为一个典型的电子商务应用，但 Magento 中是通过 block 来收集和整理网页内容。

管理 Magento 主题（Working with Magento Themes）

在这章我们将详细讲述如何管理及制作 Magento 模板主题

Magento 主题的特点

主题（Theme）这个词你可能比较熟悉了。无论是作为使用者，设计者，或两者都是，在很多 Web 应用程序中都会有主题这个选项。主题基本上两种类型的用户：首先，使用者型，在流浪商店时的美观和实用性的角度体验；第二，设计者型，就是参与主题的设计和建设的設計人員。

对于第一种用户类型，一家网店的用户体验就是能够很好的满足浏览者的视觉需求和购买需求。第二种用户类型作为创作者，最好是能够很有效率的建立一个新的网店主题。我们承认，这两个用户类型都影响到商店的最后利润，因此，即使是用户体验都不可以忽视的。

因为我们知道，作为设计师你已经设计好了漂亮的图形页面素材（为使用者服务），在这里我们只希望能够帮你建立一个强大的主题管理功能，以最大限度 地提高您的工作效率，并让你的创意到一个新的层次。下面列出的一些要点，我们认为肯定让你满意：

1. **最大化定制能力（Maximum customization power）**

通过 **Magento** 你可以在分类页面和产品页面的级别来更改页面的外观和风格，这样可以让你的网店更多样化，从而给您更强大的市场营销和促销的能力。以前是否想象过为网店的每个产品都定制独特的页面？通过 **Magento**，你可以轻松地实现这一点，而且 **Magento** 提供了快捷的方式来来自定义每个分类和每个产品的页面。

2. **多主题支持（Multiple themes）**

Magento 可以让您一次加载多个主题，允许你在默认店铺设计和临时（季节性销售等）设计间切换 — 这些操作都是非常简单的。

3. **不间断地工作（Uninterrupted workflow）**

通过 **Magento** 的完全面向对象的程序设计，所有模块都可以通过模板中的标签立即使用。而且因为 **Magento** 的功能丰富，你不必依赖程序员 来为你完成简单的工作。**Magento** 的社区成员不断成长（包括 **Magento** 的官方团体），在里面你可以找到很多的应用知识，而不用你自己摸索。

4. **最小化调试时间（Minimize debugging time）**

设计师通常会在检查工具发现错误时花费宝贵的时间检查。校验工具可能告诉你什么是错的，但是他从来不告诉你错误发生在哪里。**Magento** 的模块化的机制使得每个模板都最小化 (X)HTML 的数量，减少的数量意味着更容易发现错误所在，这样你就可以把精力更多放在真正最重要的地方。

真的，**Magento** 提供最好的地方就是它提供的灵活性，在你为网店前台定制精彩的主题时无需顾虑太多东西。天空才是 **Magento** 的极限，尽量 发挥你的想象，创造一个属于你自己的网店吧。

新建 **Magento** 主题

让我们首先介绍主题的相关一些目录。打开 **Magento** 根目录下面的目录：

- **目录 1:** `app/design/frontend/default/default/` — 此目录包含布局 (layout)，翻译 (本地) 和模板文件。
- **目录 2:** `skin/frontend/default/default/` — 此目录包含图片、CSS 和 Javascript 文件。当建立主题时，这两个目录就是文件的工作目录。

正如您可能已经注意到，我们已经将主题文件分为两部分。分开的文件可以让你的网店更加安全，**Magento** 在设计时也已经在程序的每个地方提供最高 的安全性。

让我们继续产品这两个目录。

乍一看您会发现在两个目录中的目录的名称“default/default”：

- **目录 1:** `app/design/frontend/default/default/`
- **目录 2:** `skin/frontend/default/default/`

在这两种情况下，*表示界面(Interface)的名称，并*表示主题 (Theme) 的名称。所以，如果你在建立一个界面“my_interface”下的主题“my_theme”，你的工作目录将是“app/design/frontend/my_interface/my_theme/”。

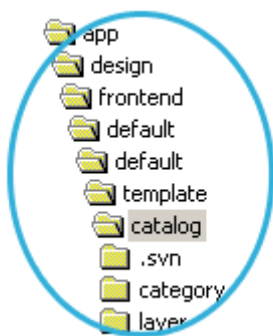
你可以在你的界面下保存多个主题，在编写本手册时，你的网店只能加载默认主题和一个另外的主题（注：Magento 的最新版已经可以加载多个主题）。要了解 Magento 是如何一次加载多个主题的，可以阅读主题的层次结构：这使用了两个主题，因此，您可以让您的网店平时使用默认主题，在季节性或事件中使用的额外的主题（要阅读有关多个主题如何工作，以及有什么好处的，阅读 Say Hello to Multiple Themes）。

创建一个新的默认主题（Creating a new default theme）

为了创建一个新的默认主题，首先您必须复制一个现有的默认主题作为新主题修改的基础。创建一个 `app/design/frontend/default/default/` 目录的副本，并重命名（界面 `interface` 和主题 `theme` 的名称应当是一个以小写字母开头的数字英文字母组合的字符串。比如 'My New Theme' 和 '02123_my_theme' 就不好，'my_new_theme' 就比较好）。程序是通过主题目录的名称来确认主题的。 `skin/frontend/default/default` 也是同样操作。就这样子！您现在已经成功地创建了一个新的默认主题。要了解如何分配这个主题 到商店上，请阅读下面的指定一个新的主题。

创建一个新的非默认主题（Creating a new non-default theme）

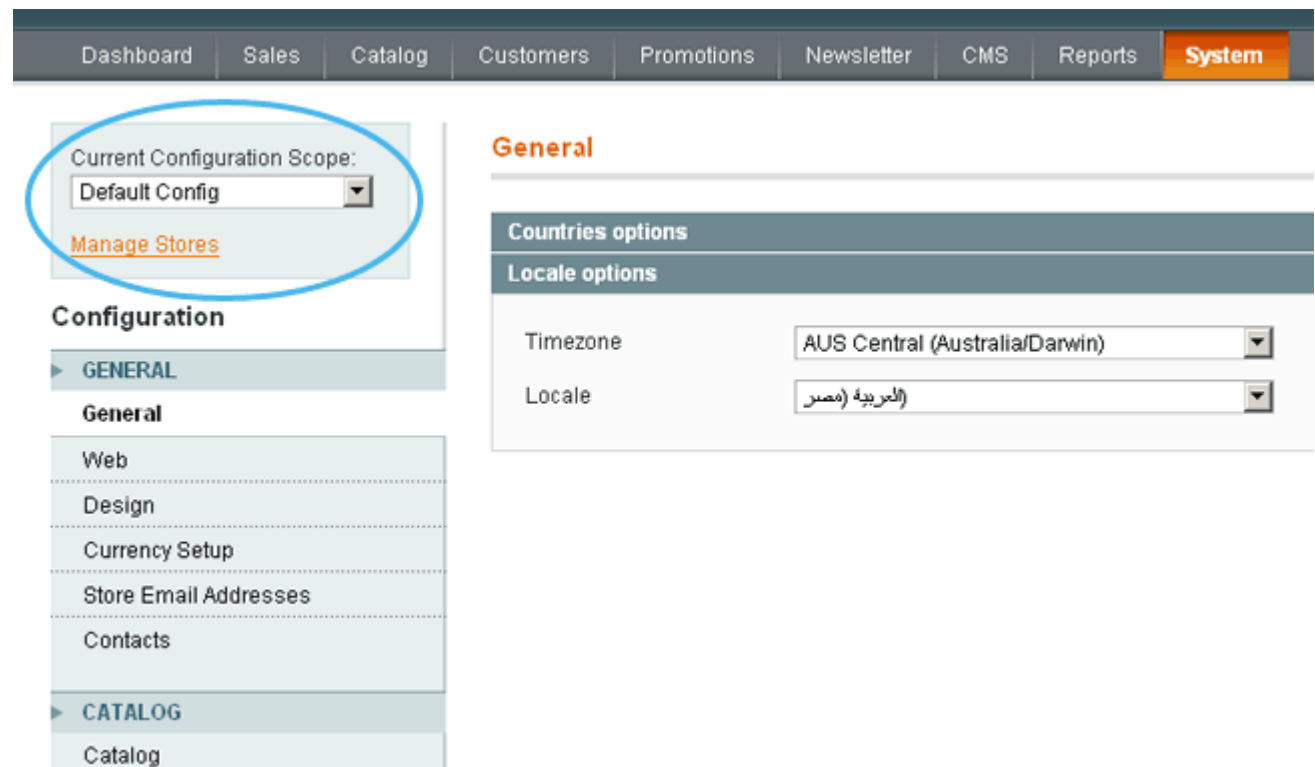
图 1



当创建一个新的非默认主题，则不需要复制任何现有的默认主题目录。最有可能的是你只要改变一些特定的文件，因此只需要复制一些需要修改的文件作为起点。有一个规则你一定要切记并遵循，你必须保持 Magento 的主题文件目录结构。例如，如果所有非默认主题在 `catalogo` 模块下都有一个文件叫 'home.phtml'，在 `app/design/frontend/your_interface/your_non_default_theme/` 目录下你需要创建一个子目录叫 'template/catalog' 存放该文件。当你打开一个默认 Magento 主题目录(图 1), 可以看到目录结构—请确保新的主题有同样的结构。

为商店指定界面和主题（Assigning interface and theme to the store）

现在，您已经创建了自己的主题（无论是默认还是非默认），您需要将分配给您的 Website/Store，以使之得以生效。转到 Magento 管理后台（ie.www.mydomain.com/admin），然后选择 Design 配置标签（System -> Configuration -> Design tab）。



在图 2 的左上角，你可以看到一个‘Current Configuration Scope’的下拉框。

- 要管理**管理站点级别的设计**，从下拉框中选择站点的名字，然后按照下面步骤操作。
- 要管理**管理 Store view 级别的设计**，从下拉框中选择 Store View 的名字，然后按照下面步骤操作。

步骤 1

在 Design 选项卡上，在 Current package name 中输入的新主题所属的界面（interface）名称。如果留空 Magento 会自动设成 ‘default’。

步骤 2

在 Default（在主题标题下面）中，输入商店要加载的此新主题的名称。如果您将此框留空，Magento 会自动加载的‘default’主题（请记住，不管您在此 Design 中如何配置，Magento 都将自动装载名字叫‘default’的主题。如果你在后台指定另外一个主题，那么它将优先被装载，但是 ‘default’的主题仍然被一同装载。在将来版本中这个策略可能被改变，不管怎样，目标都是让你能完全控制装载到店铺的主题）。你可以分别装载主题的文件类型（布局，模板，皮肤或翻译文件），相应处输入主题的名称。

步骤 3

当完成时，点击 **Save Config** 按钮，然后刷新网站—Yeah! 现在你可以在前台看到新的主题了。

现在我们已经知道如何创建和管理主题了，下面让我们学习 **Magento** 是如何处理这些主题的。

多主题入门（Say Hello to Multiple Themes）

注意:无限容量的主题将成为可与即将发布。虽然在编写本手册时时只有两个主题可以加载一次的运作幕后保持不变，您将受益于这一文件。

假期常常是网店增加销售的好机会—客户排队为他们的家人和朋友购买圣诞礼物，妈妈排队为他们的孩子买万圣节服装。为了鼓励顾客光顾你的店铺，你必须 仔细调整店铺设计以适应季节性要求。像图 3 所示的店铺没有在圣诞节气氛 — 其实这时的网店可能就需要一点点红色添加吸气，一点雪花和圣诞老人的形象 — 如图 4 做的就比较好!

图 3：未经修改的设计

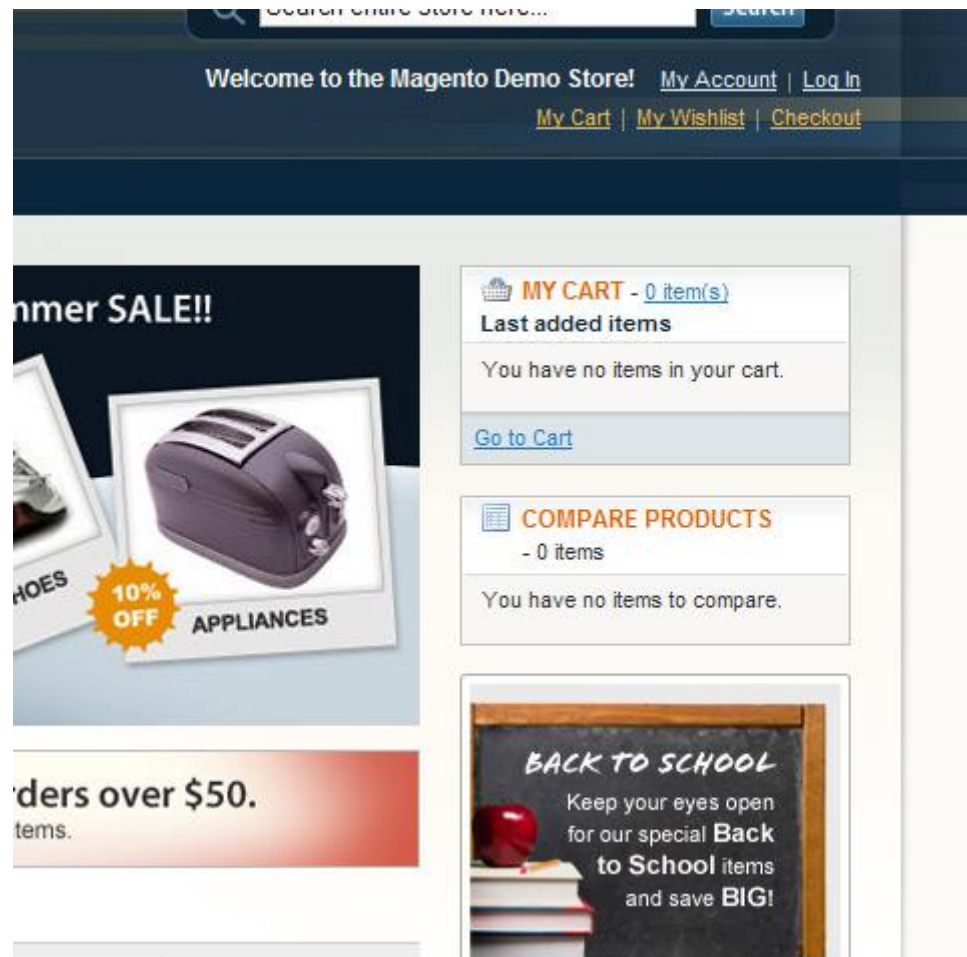
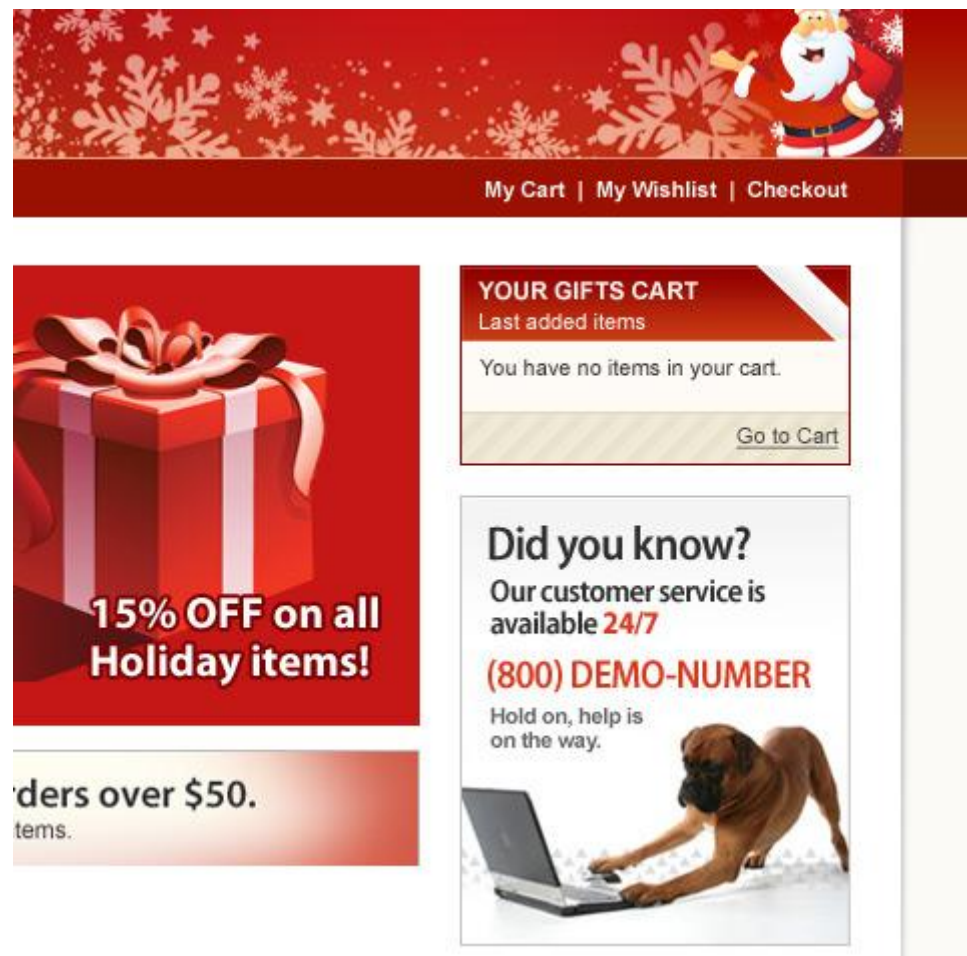


图 4: 你需要的设计



通过 **Magento** 你可以为商店处理多个主题的能力，以便在不同的时候使用。在店铺载入多个主题，你可以在平常使用普通的店面设计，在圣诞时就可以 使用圣诞主题。

通过对上面的两个设计的仔细查看，您会发现他们有相似的基本设计。虽然图 4 是圣诞主题，但是还可以看到是普通主题的页面结构作为基础。两者之间唯一 真正有区别的是设计方面的，仅仅可能包括少数几个 **CSS** 和图像文件和稍微改变的模板文件。由于实际上变化是轻微的，你不需要一个全新的默认主题来适应您的 圣诞主题。您所需要的仅仅是替换几个文件，让你的商店更加好更加有趣。**Magento** 的多个主题功能正是为此需要建立的，你只需要点击一下季节性主题的开 关就可以实现改变，同时也可以维护您的默认主题。

Magento 处理装载多个主题指派所谓层次的主题仅仅是一个过程取消了多余的文件加载只有那些居住在 **hierarchy.The** 最高层次是由您在 管理的 设计和配置选项卡你有能力这样做因此将成为功能齐全即将稳定的版本。在撰写本文时，该等级的主题是已经决定的你，作为 **Magento** 加载的默认'主题在系 统第一次（将它放在最低等级） ，然后载入中的第二个主题您指定在管理（将其置于在最高等级）。

Magento 通过主题被指定的层次（**Hierarchy**）来处理装载的多个主题，这仅仅是一个取消在高层次的主题中已经存在的文件的多余的文件加 载。层次可以在后台的 **Design** 标签中指定。在撰写本文时，主题的层次已经决定， **Magento** 首先加载'default' 主题（将它放在最低层次）， 然后载入你在后台中指定的第二个主题（将其置于在最高层次）。

主题的层次（**Hierarchy of Themes**）

当您在 **Magento** 中为店铺指定多个主题时，实际上你已经用到了这个层次（**Hierarchy**）的好处。创建主题是为了生成令人喜爱的图形界面，那么 **Magento** 的目标就是确保应用能够定位和正确 无误地装载要求的主题，保证应用程序运行不出现错误。

例如，如果您的分类列表页面调用'**view.phtml**'（在这种情况下，此模板成为一个必需的文件），但是程序在高层次的主题中无法找到该文件 （注：在撰写本手册时，最高层次主题是你后台指定的，最低层次的主题是 **Magento** 自动加载的'**default**'主题。在讲来发布的版本 中，**Magent** 将为您提供能够完全控制您的主题层次的功能），它会在下一层次的主题中寻找该文件。如果这种失败，它将继续努力降低主题的层次寻找，直至 它能够找到'**view.phtml**'文件时，该就停止搜索并加载。这种方法的建筑设计被称为 **fallbacks** ，因为应用程序'**fall back**'下一个所需文件的可能来源，以便检索和加载它。

假如你为店铺指定了三个主题，每个主题包含的文件如下：

表格 1

default	my_theme_1	my_theme_2
All required files	templates/3-col-layout.phtml	templates/3-col-ayout.phtml
	templates/header.phtml	css/base.css
	images/logo.gif	
	css/base.css	
	css/boxes.css	

这三个主题的层次如下：

表格 2

HIGHEST	my_theme_2
	my_theme_1

LOWEST	default
---------------	---------

仔细观察，在表 1 有几个冗余的文件如 `templates/3-col-layout.phtml` 和 `css/base.css`。现在让我们重新排表格让这些冗余文件在主题之间平分排列。

表格 3

default	my_theme_1	my_theme_2
All required files		
	<code>templates/3-col-layout.phtml</code>	<code>templates/3-col-layout.phtml</code>
	<code>templates/header.phtml</code>	
	<code>images/logo.gif</code>	
	<code>css/base.css</code>	<code>css/base.css</code>
	<code>css/boxes.css</code>	

你可能会问：“Ok, 太好了.但是这意味着什么？”
好，让我提醒你，表 3 中的那些文件是你看这些文件的方式，而不是 **Magento** 看这些文件的方式。

表 4 告诉你 **Magento** 怎样看待这些文件的装载：

表格 4

default	my_theme_1	my_theme_2
----------------	-------------------	-------------------

All required files		
		templates/3-col-layout.phtml
	templates/header.phtml	
	images/logo.gif	
		css/base.css
	css/boxes.css	

看到了吗？**Magento** 是怎样忽略层次里那些冗余文件和仅仅识别层次里更高的主题的文件。这是因为当它发现要求的文件就中止查找层次低的主题中的该文件而继续查找还没有发现的文件。

创建自己的主题（Building Your Theme）

Magento 是建立在完全模块化的模式基础上的，这为你的网店带来无限的可扩展性和灵活性。在建立主题时这种方式就会显现出来。在本章中，我们将探讨这意味着什么，并教会你如何为你的 **Magento** 店铺建立你自己的主题。

在 **Magento** 之前你很可能已经使用过其他电子商务程序。可能你在开发网店主题方面有一定的经验了，但是我们希望你抛弃以往经验的来的所有期待，这并不意味着你要学习一个全新的语言，也不意味着你要改变你的工作流程，只是你要学习一些新的技巧来开发 **Magento** 的主题。掌握这些工具并且始终注意关注他们，你会喜欢上这种结构的。准备好了吗？下面我们开始介绍：

1. 结构区块（Structural Blocks）
2. 内容区块（Content Blocks）
3. 布局（Layout）

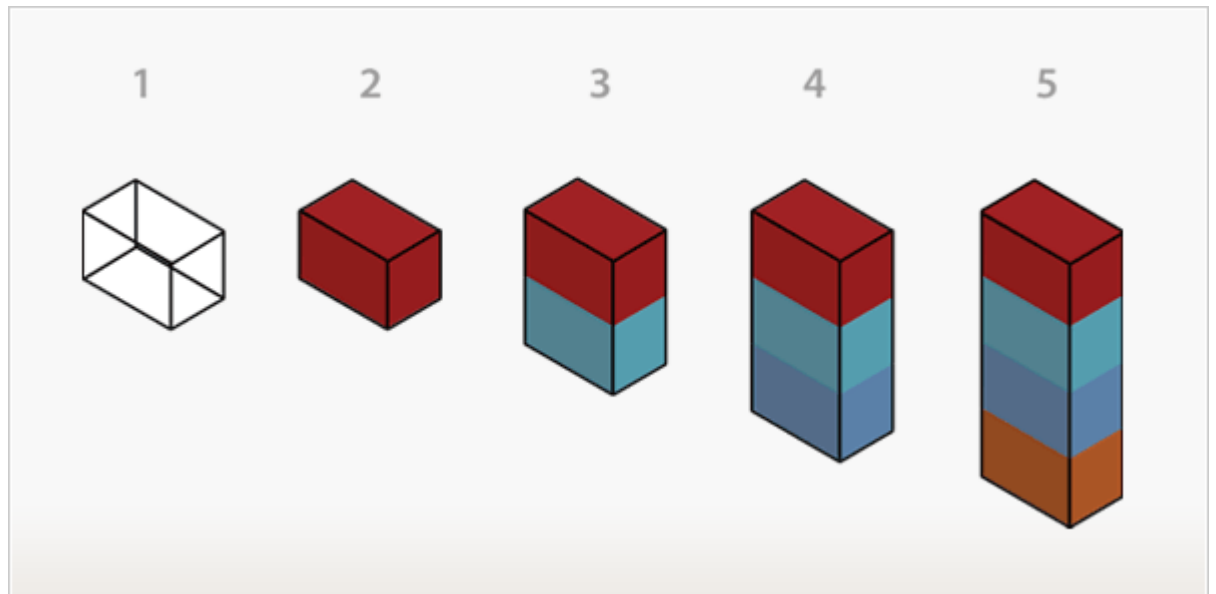
建立心理导向图（Creating the Mental Note）

为了让您清楚地了解什么是块（block）和布局（layout），这里有一个心理图片，您可以借鉴一下：

1. 想象一个长方体的轮廓(如图 1 – 不要一次性浏览整个图片，顺着下面想象下去)
2. 想象整个轮廓被填满。

3. 想象现在是两个块，他们的轮廓叠在一起。
4. 想象现在是三个块，他们的轮廓叠在一起。
5. 现在是四个块，他们的轮廓叠在一起。

图 1



6. 现在我们来查看一些一个分类页面的缩略图

图 2



7. 现在看看上面的缩略图被分割的两种情况快照

图 3



图 4



刚才建立的思维导向图给为你在概念和真实区块之间进行了一个平行的比较，也许你完全糊涂了，让我来解释一下。

从概念角度看，图 3 中的框架是结构区块。他们是内容区块的父区块，他们帮助定义内容区块在网店页面中的显示位置（如在图 4 中）。这些结构区块通常的存在形式为页眉区，左侧栏区，右侧栏区...等等。这有助于创建网店的视觉结构。

另一方面，内容区块概念上就是装点结构区块的独立颜色的区块。在一个网店内容里，他们才是真正的内容。内容区块代表着页面内的每一个功能特性（比如 分类列表，标注和产品标签...等等），并且使用模板文件生成(x)HTML 插入到父结构区块里。

布局（Layout） 工具可以为每个你创建的结构区块分配内容区块。布局是以 XML 文本文件的形式存在，可以修改各个页面中各个结构区块的设计和指定显示模板的内容区块。事实上，通过一些布局文件的帮助，你可以修改网店每一个页面的视觉效果。要了解更多阅读介绍布局（Intro to Layouts）一章。

在 Magento 里你不再需要 `left_column.ext` 这个模板文件，取而代之的是通过管理每个功能性基础元素来管理模板。你可以通过一些布局命令载入或者卸载功能元素来控制网店页面。

下面是 Magento 提供的建立主题的工具：

1. **模板（Templates）**
2. **布局（Layouts）**
3. **区块（Blocks）**
4. **皮肤（Skins）** (images, CSS and block-specific Javascript)

要为 Magento 网店建立一个主题，下面是你要操作的其中一些步骤：

第一步：关闭系统缓存(Disable your system cache)

为了让 Magento 系统执行一个完整的过程，需要在后台先禁用系统缓存（<http://yourhost.com/admin>）和 System -> Cache Management。在 All Cache 中选择 Disable，然后点击保存。这样做将确保您可以在网店中看到实时更改的反映。

第二步：确定你网店可能使用的所有结构类型

甚至在你开始创建的商店的标记之前，您首先需要问一下自己要为你的网店的每个页面结构定义好一个页面类型。你自己可能会列一个类似于下面的清单：

- 主页将使用三栏结构 *three column structure*。
- 分类列表页面使用两栏结构并带一个有侧栏 *two column structure that includes a right column*。
- 客户页面使用两栏结构并带一个左侧栏 *two column structure that includes a left column*。

骨架模板（Skeleton template）

当你完成了这个清单之后，你会为每个结构类型建立(x)HTML 并在

`app/design/frontend/your_interface/your_theme/template/page/`下 保存这些骨架模板。一个骨架模板就是包含一个页面的总体结构（除的块），是为每个结构区块定义好了总体结构。

骨架模板例子:

```
<html>
<head></head>
<body>
<div class="header"><?=$this->getChildHtml('header')?></div>
<div class="middle">
    <div class="col-left"><?=$this->getChildHtml('left')?></div>
    <div class="col-main"><?=$this->getChildHtml('content')?></div>
</div>
<div class="footer"><?=$this->getChildHtml('footer')?></div>
</body>
</html>
```

浏览上面的简单骨架模板, 你会注意到每个表现的标记中有一个 PHP 方法 `getChildHtml()`。这就是 **Magento** 将结构区块加载到骨架模板中的方法, 并且决定了内容区块在商店页面中的显示位置。

每个 `getChildHtml` 都会调用一个名称 `<div class="header"><?=$this->getChildHtml('header')?></div>`, 这些名称是在布局中结构区块的唯一标识。骨架模板通过布局应用到商店中。

第三步： 根据功能性修剪你的(X)HTML

一旦您建立您的骨架模板, 您现在需要创建的模板每个内容块。

Magento 喜欢有意义模板

您将需要削减你的页面的(X)HTML 标记, 并根据每个标记为页面提供功能。举例来说, 如果你有 **mini-cart** 区的设计, 此区域的标记将有助于 这个区域的自己的模板文件。同时产品标签, **newsletter**, 注册等...所有这些功能 **Magento** 都已经配备了, 因此您可以参考现有的模板标记来 建立您的标记逻辑。

哪里保存模板

图 5



商店中的每个页面的所有标记（包含一系列的模板）都是表示一个模块的功能。当你想修改一个页面调用的模板，可以他开 **Template Path** 提示。要打开这个功能，按下面操作：

1. 进入后台，到 **System -> Configuration**
2. 在右上角的 **Website/Store** 选择器中选择你的商店
3. 然后选择左侧栏的 **Developer** 标签，并在 **Template Path Hints** 中选择 **Yes**

当您完成后，回到商店前台，重新加载您的网页，您将会看到所有模板块的路径。要更改标记的话，根据显示的路径找到并修改响应的模板。

第四步：改变布局以适应你的设计

一旦您修改一些标记，你可能想移动一下这些模板看看这些模板修改的效果是什么样的。

如何找到布局文件

要访问布局文件，可以在 `app/design/frontend/your_interface/your_theme/layout/` 找到。就像模板文件一样，布局文件也是基于每个模块的方式存储的，这样你就可以方便的按模板提示来进行修改。首先，激活模板提示，刷新要修改的页面，找寻模板提示的模板路径。比如 你想移动 **mini cart**，参考模板路径(例如: `app/design/frontend/default/default/checkout/cart/sidebar.phtml`)，用主题文件夹里第一个文件夹名(字体加重的那个)来寻找相关布局文件。在这个例子里，我们要寻找 '`checkout.xml`' 来修改 **mini cart** 的位置。

默认布局（Default Layout）和布局更新（Layout Updates）

一共有两种布局：默认（default）和更新（updates）。一个默认布局(例如 `page.xml`)是默认应用到网店每个页面的布局。所有其他的布局文件都是更新布局，也就是在默认布局基础上进行响应的更新布局。

让我们举例说明一下骨架模板（skeleton template）:

在默认布局里，你已经设置为三栏结构，就是说，默认几乎所有的页面都是三栏结构。但是在产品页面并不需要三栏，对产品

页面来说，你需要一个包括右侧栏的两栏结构布局。为了实现这一点，你不用理默认布局，只要打开 `catalog.xml` 文件，你可以在里面修改一些你要实现包含右侧栏的两栏布局命令。这就是所谓更新了一个布局（`updating a layout`）。

```
<reference name="root">
```

```
    <action  
method="setTemplate"><template>page/2columns-right.phtml</template></  
action>
```

```
</reference>
```

让我们看看另外一个例子：

默认情况下，你想订阅 `newsletter` 区块显示在有侧栏，但在客户的帐户页面不显示它。在这种情况下，你可以需要打开 `customer.xml` 文件，在里面你可以删除了 `newsletter` 内容区块，这就让此页面不包括此 `newsletter` 功能。

布局介绍（Intro to Layouts）

布局（`Layout`），由其组建的表现形式来说很容易让你以为要使用它你首先必须学习广泛的编程逻辑知识。但是事实上，布局只是由一些很容易学习的 `XML` 标记组成的。通过学习一些关键的概念和布局的命令，你很快就会建立了相关的信心和知识，以方便地为你的店铺设计结构。

图 1:

```
<default> → Handle
<!-- Mage_Catalog -->
<reference name="top.menu">
    <block type="catalog/navigation"
</reference>
<reference name="left">
    <block type="core/template" name
        <action method="setImgSrc"><
        <action method="setImgAlt"><
        <action method="setLinkUrl">
        <action method="setLinkUrl">
    </block>
</reference>
<reference name="right">
    <block type="core/template" name
    <block type="core/template" name
</reference>
</default>

<!--
Category.default.layout
-->

<catalog_category_default> → Handle
<reference name="left">
    <block type="catalog/navigation"
</reference>
<reference name="content">
    <block type="catalog/category_vi
        <block type="catalog/product
    </block>
</reference>
</catalog_category_default>
```

布局(Layout)是 Magento 应用中的一个虚拟组件。通过修改了部分的布局，您可以为店铺页面建立升级兼容的页面修改。

布局是由默认布局和更新布局组成，它们都是一些易于学习 XML 标记。通过这些布局的命令，您可以修改/指定的内容区块和结构区块的关系，并控制前台 店铺的功能，例如可以在具体的页面中加载和不加载的具体 Javascripts 文件。

布局文件是根据模块分开的，每一个模块带来了自己的布局文件(例如 'catalog.xml' 是一个分类模块的布局文件，'customer.xml' 是客户模块的布局文件...等)。这些布局文件位于

app/design/frontend/your_interface/your_theme/layout/ 中和每个文件是进一步分开处理（见图 1），每个处理（除了<default>）指定其在店铺特定页面的嵌套更新。

一些布局文件中可能包含了<default>处理。在解析布局文件时，Magento 首先寻找分配到<default>的更新布局处理几乎所有的布局文件，读取他们的命令，在 app/etc/modules/Mage_All.xml 中 指定。然后解析特定页面布局更新，最后建立一个商店的页面。

Magento 系统使用这种方式，以便无缝添加和删除模块而不影响其它模块的系统。

布局原理解析（Anatomy of Layout）

Magento 中的布局（Layout）包含一小部分的标记集合，作为详细说明关于程序如何建立一个页面，如何建立它的行为和每个构建的区块。最佳的布局途径是在每个角度正确的划分和使用。为了让您能够做到这一点，下面是一些行为特性的布局 XML 标记。

句柄（Handle）

Handle (图 1)是一个标识符，决定应用程序要如何通过嵌套的更新处理它。

如果句柄的名称是<default>，然后应用程序知道在加载网店的几乎所有页面之前应该加载此特定页面布局的嵌套更新（我们说'几乎所有的'，因为一些特殊的页面像产品图片弹出窗口就没有加载布局中的<default>句柄）。

如果 Magento 找到<default>以外的句柄，它将按照指定的句柄中的页面嵌套更新对页面进行处理。例如，<catalog_product_view>包含 Product View 页面的布局更新，而<catalog_product_compare_index>包含 Compare Product 页面的更新布局。句柄作为设计人员设置在网店中的标识符，他不需要广泛的理解 Magento 编程，也不应该需要修改。

<block>

Magento 通过<block>标记决定页面中的每个区块的行为和视觉表现。在 Magento 中我们已经提到了两种类型的区块-结构区块（structural blocks）和内容区块（content blocks）。区分这两种区块最好的方式是通过分配给它的标记属性来区分。结构区块通常包含属性'as'，通过这个属性值程序可以与指定的区域（由 getChildHtml 方法指定）中的模板联系。你会发现在默认布局许多地方出现这个'as'属性，因为默认布局的一个性质就是建立一个实际的布局，在各个不同的页面中的具体布局上就可以开始增加。例如，在默认布局中，有像'left'、'right'、'content'和'footer'这些结构区块。并不是说这些区块不能存在于正常的布局更新中，但我们为什么不首先在默认布局中建立这些结构区块，然后在后面每个具体的页面基础上添加内容呢？让我们进一步挖掘<block>的现有属性。

- **type** – 这是模块类的标识符，它定义了区块的功能。此属性不应该被修改。
- **name** – 这是名称，其他的区块可以通过此名称引用此区块（看图 3）。
- **before (and) after** – 这两种方法决定内容区块在结构区块中的位置。before="." 和 after="."这样的命令标志此区块的位置是一个结构区块的最上方或最下方。
- **template** - 这个属性指定的值决定了此区块的功能是使用哪个模板。例如，如果这个属性值指定了'catalog/category/view.phtml'，程序就会载入'app/design/frontend/template/catalog/category/view.phtml' 模板文件。要了解布局是如何通过模板执行的，阅读分步指南建设一个主题。
- **action** – <action> 是用来控制前台的功能的，如加载或不加载一个 JavaScript。一套完整的 action 方式将很快推出，但此时的最佳的学习途径是了解现有的布局更新上面的不同 Action 方法。

- **as** – 此属性指定模板文件中会调用那个区块。当您在模板中看到 `getChildHtml(' block_name ')` 的 PHP 方法，可以肯定它指的是引用属性 `as` 的值为 `' block_name '` 的区块。（例如：在骨架模板中的方法 `<?=$this->getChildHtml('header')?>` 是调用 `<block as="header">`）

<reference>

`<reference>` 是用来引用另一个区块。要引用灵位一个区块，在内部的更新将应用于与其关联的 `<block>`（见图 3）。

要使用引用，可以通过区块中的 `'name'` 属性值引用。此属性的指向标签中 `'name'` 属性。所以，如果你使用 `<reference name="right">`，响应的区块名称将是 `<block name="right">`。

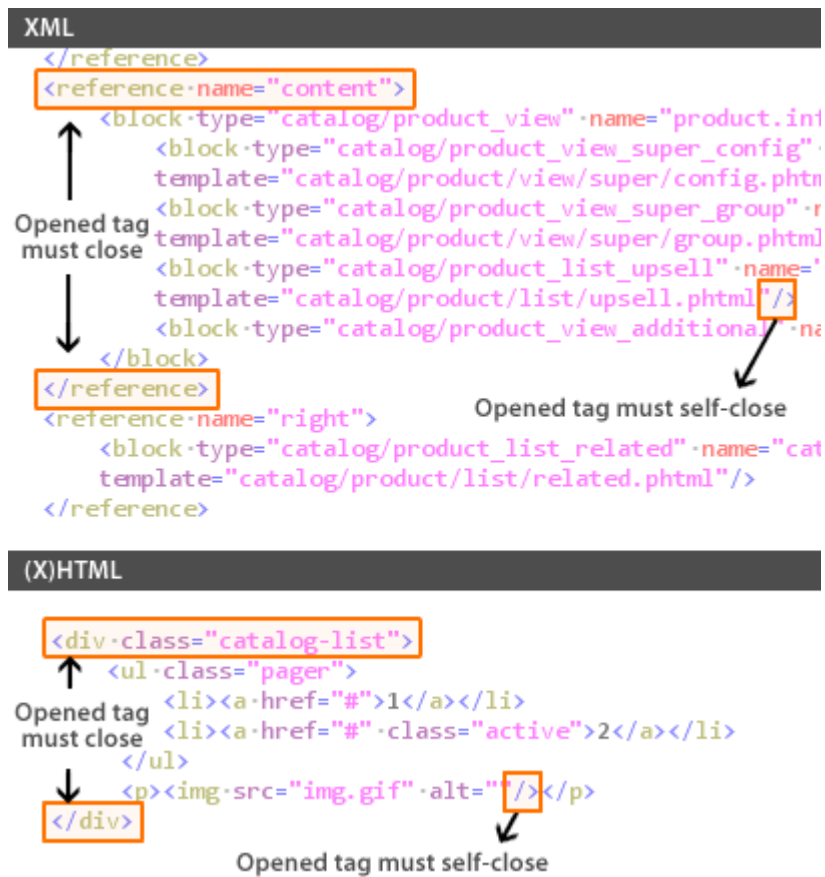
图 3:



XML 的规则（Rules of XML）

如果你不熟悉 XML，在遇到 Magento 中基于 XML 的布局更新，您可能会对此规则有一两个问题，您必须按照此规则修改 XML 文件。你要记得关于 XML 的是，当一个标签打开后，它必须继之以一个结束标记（`<xml_tag></xml_tag>`）或自我结束（`<xml_tag/>`），跟(X) HTML 文件的标签一样。

图 4:



快速学习的例子 (Quick Exercises to Get You Started)

就像其他新的知识，最好的学习方法就是通过例子来学习。为了你能够快速的学会使用布局，下面准备了两个 Magento 的联系，在进行此例子之前你要 准备好 Magento 的默认主题。

要学习这两个例子，可以看这里

到此你已经阅读完了 Magento 设计人员手册。我们希望你已经学习到了设计 Magento 模板系统的 相关知识。我们也希望在本站的 [Magento 论坛](#)中 看到你的学习经验交流，在论坛中你可以跟社区的其他成员一起分享 Magento 的相关经验，欢迎你的加入。